

Docket No. AUS920010083US1

**METHOD AND APPARATUS FOR HANDLING REQUESTS FOR CONTENT IN
A NETWORK DATA PROCESSING SYSTEM**

BACKGROUND OF THE INVENTION

5

1. Technical Field:

The present invention relates generally to an improved data processing system, and in particular to a method and apparatus for handling a request for content.

10 Still more particularly, the present invention provides a method, apparatus, and computer implemented instructions for handling a request for content, which may originate from different content providers.

15 **2. Description of Related Art:**

The Internet, also referred to as an "internetwork", is a set of computer networks, possibly dissimilar, joined together by means of gateways that handle data transfer and the conversion of messages from protocols of the
20 sending network to the protocols used by the receiving network (with packets if necessary). When capitalized, the term "Internet" refers to the collection of networks and gateways that use the TCP/IP suite of protocols.

The Internet has become a cultural fixture as a
25 source of both information and entertainment. Many businesses are creating Internet sites as an integral part of their marketing efforts, informing consumers of the products or services offered by the business or providing other information seeking to engender brand loyalty. Many
30 federal, state, and local government agencies are also employing Internet sites for informational purposes, particularly agencies which must interact with virtually

Docket No. AUS920010083US1

all segments of society such as the Internal Revenue Service and secretaries of state. Providing informational guides and/or searchable databases of online public records may reduce operating costs. Further, the Internet
5 is becoming increasingly popular as a medium for commercial transactions.

Currently, the most commonly employed method of transferring data over the Internet is to employ the World Wide Web environment, also called simply "the Web". Other
10 Internet resources exist for transferring information, such as File Transfer Protocol (FTP) and Gopher, but have not achieved the popularity of the Web. In the Web environment, servers and clients effect data transaction using the Hypertext Transfer Protocol (HTTP), a known
15 protocol for handling the transfer of various data files (e.g., text, still graphic images, audio, motion video, etc.). The information in various data files are formatted for presentation to a user by a standard page description language, the Hypertext Markup Language
20 (HTML). In addition to basic presentation formatting, HTML allows developers to specify "links" to other Web resources identified by a Uniform Resource Locator (URL). A URL is a special syntax identifier defining a communications path to specific information. Each logical
25 block of information accessible to a client, called a "page" or a "Web page", is identified by a URL. The URL provides a universal, consistent method for finding and accessing this information, not necessarily for the user, but mostly for the user's Web "browser". A browser is a
30 program capable of submitting a request for information identified by an identifier, such as, for example, a URL. A user may enter a domain name through a graphical user interface (GUI) for the browser to access a source of

Docket No. AUS920010083US1

content. The domain name is automatically converted to the Internet Protocol (IP) address by a domain name system (DNS), which is a service that translates the symbolic name entered by the user into an IP address by looking up
5 the domain name in a database.

Currently, requests for content are handled by services designed for each particular type of content source. For example, Domino provides integrated application services-such as security, workflow and
10 content management-optimize the platform for rapid delivery of the collaborative Web applications you need to initiate and strengthen key business relationships. Domino is available from International Business Machines Corporation (IBM). Another content source may employ an
15 Internet Information Server (IIS), which is a Web server system available from Microsoft Corporation. Currently, each of these types of content sources require a separate set of processes to process requests and deliver content, such as Web pages. Each service also may provide
20 features, such as electronic mail and security. Many times, a customer may use a different type of content source to serve or provide Web applications. The customer often requires a consistent look and feel regardless of the type of content source. This
25 requirement often involves duplication of common services and infrastructure. When a customer has more than one type of content source, such as a Domino system and a DB2 system, changes made to the look and feel within one content source system require similar changes to be made
30 to other content source system.

Further, often times a customer will have different developers create or modify content. It is often difficult a desired presentation with different

Docket No. AUS920010083US1

developers handling the content creation and modification. Currently, the content has to be reviewed and changes may have to be made to maintain a consistent presentation of content from the content source.

- 5 Therefore, it would be advantageous to have an improved method and apparatus for maintaining a desired presentation of content from multiple content sources.

Docket No. AUS920010083US1

SUMMARY OF THE INVENTION

The present invention provides a method, apparatus,
5 and computer implemented instructions for handling a
request for content in a data processing system. The
request for content is received. A content source is
identified from a plurality of content sources using the
request to form an identified content source. The
10 content is retrieved from the identified content source.
A template is applied to the content to generate a web
page, wherein the template is a template associated with
the identified content source from the plurality of
content sources.

BRIEF DESCRIPTION OF THE DRAWINGS

The novel features believed characteristic of the
5 invention are set forth in the appended claims. The
invention itself, however, as well as a preferred mode of
use, further objectives and advantages thereof, will best
be understood by reference to the following detailed
description of an illustrative embodiment when read in
10 conjunction with the accompanying drawings, wherein:

Figure 1 is a pictorial representation of a network
of data processing systems in which the present invention
may be implemented;

Figure 2 is a block diagram of a server cluster in
15 accordance with a preferred embodiment of the present
invention;

Figure 3 is a block diagram of a data processing
system that may be implemented as a server in accordance
with a preferred embodiment of the present invention;

Figure 4 is a diagram of a content server system in
20 accordance with a preferred embodiment of the present
invention;

Figure 5 is a diagram of tiers within a content
server system in accordance with a preferred embodiment
25 of the present invention;

Figure 6 is diagram of data flow used to generate a
Web page in accordance with a preferred embodiment of the
present invention;

Figure 7 is a diagram of table used to identify
30 content sources in accordance with a preferred embodiment
of the present invention; and

Docket No. AUS920010083US1

Figure 8 is a flowchart of a process used for processing a request for content in accordance with a preferred embodiment of the present invention.

Figure 8 is a flowchart of a process used for processing a request for content in accordance with a preferred embodiment of the present invention.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

5 With reference now to the figures, and in particular
Figure 1 depicts a pictorial representation of a network
of data processing systems in which the present invention
may be implemented. Network data processing system 100 is
a network of computers in which the present invention may
10 be implemented. Network data processing system 100
contains a network 102, which is the medium used to
provide communications links between various devices and
computers connected together within network data
processing system 100. Network 102 may include
15 connections, such as wire, wireless communication links,
or fiber optic cables.

In the depicted example, a server 104 is connected
to network 102 along with storage unit 106. Further,
network data processing system 100 also includes server
20 cluster 108 as well as a server 110. In this example,
server 110 is connected to network 102 through local area
network (LAN) 112. In addition, clients 114, 116, and 118
also are connected to network 102. These clients 114,
116, and 118 may be, for example, personal computers or
25 network computers. In the depicted example, server 104,
server cluster 108, and server 110 provide a distributed
web commerce server system for processing requests from
clients 114-118. In the depicted examples, server 104,
server cluster 108 and server 110 may be located in
30 multiple, geographically distributed sites. The servers
are presented to users, such as those at clients 114-118,
as a single Web site. Additionally, one of the servers,

Docket No. AUS920010083US1

such as server 104 may act as a load balancer to receive and direct requests from the clients to the appropriate servers within the system. Network data processing system 100 may include additional servers, clients, and
5 other devices not shown.

In the depicted examples, network data processing system 100 is the Internet with network 102 representing a worldwide collection of networks and gateways that use the TCP/IP suite of protocols to communicate with one
10 another. At the heart of the Internet is a backbone of high-speed data communication lines between major nodes or host computers, consisting of thousands of commercial, government, educational and other computer systems that route data and messages. Of course, network data
15 processing system 100 also may be implemented as a number of different types of networks, such as for example, an intranet, a local area network (LAN), or a wide area network (WAN). **Figure 1** is intended as an example, and not as an architectural limitation for the present
20 invention.

With reference now to **Figure 2**, a block diagram of a server cluster is depicted in accordance with a preferred embodiment of the present invention. Server cluster 200
in this example may be implemented as server cluster 108
25 in **Figure 1**.

In this example, servers 202-210 are in communication with each other through communications system 212, which may take various forms. Communications system 212 may be, for example, a bus, a network, or a
30 shared memory. Communications system 212 is used to handle routing of requests and responses directed towards server farm 200. Load manager 214 also is connected to

Docket No. AUS920010083US1

communications system 212 and serves to receive requests directed to server farm 200 from network 216. Load manager 214 also serves to distribute requests to servers 202-210 for processing.

5 Referring to Figure 3, a block diagram of a data processing system that may be implemented as a server, such as server 104 in Figure 1, is depicted in accordance with a preferred embodiment of the present invention. Further, data processing system 300 may be implemented as
10 a server, such as server 202 with server cluster 200, in Figure 2.

Data processing system 300 may be a symmetric multiprocessor (SMP) system including a plurality of processors 302 and 304 connected to system bus 306.
15 Alternatively, a single processor system may be employed. Also connected to system bus 306 is memory controller/cache 308, which provides an interface to local memory 309. I/O bus bridge 310 is connected to system bus 306 and provides an interface to I/O bus 312.
20 Memory controller/cache 308 and I/O bus bridge 310 may be integrated as depicted. Peripheral component interconnect (PCI) bus bridge 314 connected to I/O bus 312 provides an interface to PCI local bus 316. A number of modems may be connected to PCI bus 316. Typical PCI
25 bus implementations will support four PCI expansion slots or add-in connectors. Communications links to network computers 114-118 in Figure 1 may be provided through modem 318 and network adapter 320 connected to PCI local bus 316 through add-in boards.

30 Additional PCI bus bridges 322 and 324 provide interfaces for additional PCI buses 326 and 328, from

Docket No. AUS920010083US1

which additional modems or network adapters may be supported. In this manner, data processing system 300 allows connections to multiple network computers. A memory-mapped graphics adapter 330 and hard disk 332 may
5 also be connected to I/O bus 312 as depicted, either directly or indirectly.

Those of ordinary skill in the art will appreciate that the hardware depicted in **Figure 3** may vary. For example, other peripheral devices, such as optical disk
10 drives and the like, also may be used in addition to or in place of the hardware depicted. The depicted example is not meant to imply architectural limitations with respect to the present invention.

The data processing system depicted in **Figure 3** may
15 be, for example, an IBM RISC/System 6000 system, a product of International Business Machines Corporation in Armonk, New York, running the Advanced Interactive Executive (AIX) operating system.

With reference now to **Figure 4**, a diagram of a
20 content server system is depicted in accordance with a preferred embodiment of the present invention. In this example, content server system 400 may be implemented within a server system, such as server farm 200 in **Figure 2** or data processing system 300 in **Figure 3**. In these
25 examples, the process, navigation, and display components reside within a Java application server.

In the depicted examples, server system 400 includes page creation process 402, which receives user request
404. As illustrated, user request 404 is passed using
30 the HTTP standard. The mechanism of the present invention may be applied to any content sources supporting this standard.

Docket No. AUS920010083US1

In response to receiving request 404, page creation 402 will obtain navigation information from navigation database 406 using a universal resource identifier (URI) within the request. Navigation database 406 provides content used to generate links or other mechanisms used to navigate to different Web pages. In these examples, the URI is in the form of a universal resource locator (URL). DataBase 2 (DB2) is an example of a database system, which may be used to implement navigation database 406. DB2 is a relational database management system available from IBM. DB2 is a full-featured SQL language database management system.

Page creation process 402 also will query either content source 408 or content source 410 for content using the URL within user request 404. As illustrated, content source 408 and content 410 provide content for Web pages. In the depicted examples, content source 408 may be a Domino server, while content source 410 may be an IIS server. In response to receiving content and navigation information, Web page 412 is generated and returned to the user.

With this system, a centralized control of page presentation is made possible. Page creation process 402 controls the presentation of content from multiple content sources. In this manner, developers creating content are no longer required to interface with developers generating navigation information. Further, the addition of other content sources, do not require changes to navigation database 406, content source 408, or content source 410. These changes only need to be made to page creation process 402. In this manner, scalability is provided for adding content sources and

Docket No. AUS920010083US1

navigation information, without requiring changes to the different content services.

Turning next to **Figure 5**, a diagram of tiers within a content server system is depicted in accordance with a preferred embodiment of the present invention. In the depicted examples, content server system 500 includes data layer 502, application logic layer 504, and presentation layer 506. Data layer 502 contains information sources, such as navigation database 508, content source 510, and content source 512. Application logic layer 504 includes request process 514, get navigation process 516, get content process 518, and build page process 520, which may be implemented as page creation process 402 in **Figure 4**.

Request process 514 serves to receive requests from users, such as a request for a Web page from a browser program. In response to receiving a request, request process 514 identifies a URL sent within the request. This URL is sent to get navigation process 516 and get content process 518. In turn, get navigation process 516 will request navigation data from navigation database 508 using the URL. Get content process 518 will request content from content source 510 or content source 512 using the URL. The requested information is stored in temporary data 522, which may be, for example, a portion of memory or space within a hard disk drive.

Request process 514 also will initiate build page process 520. Build page process 520 will generate a Web page, such as Java server page (JSP) 524 in presentation layer 506. A JSP is an HTML page with embedded Java source code that is executed in the Web server or

Docket No. AUS920010083US1

application server. The HTML provides the page layout that will be returned to the Web browser, and the Java provides the processing, for example, to deliver a query to the database and fill in the blank fields with the results. JSP 524 contains navigation area 526 and content area 528. Navigation information is located in navigation area 526, while content is placed in content area 528. The presentation or look and feel of JSP 524 is controlled by build page process 520.

Turning now to Figure 6, a diagram of data flow used to generate a Web page is depicted in accordance with a preferred embodiment of the present invention. The data flow in Figure 6 may be implemented within page creation process 402 in Figure 4.

The data flow begins by receiving user request 600. Request process 514 uses the request information from user request 600 to get/create session information from session database 602 (block S1). A user object is then created (block S2) and stored within temporary data 522. Get navigation process 516 is initiated to query for navigation information from DB2 604, which contains the navigation information (block S3). This query is made using the URL obtained from user request 600. The navigation object is then created (block S4) and stored with temporary data 522. Get content process 518 uses the URL to obtain content from an application at server 606 supporting HTTP in these examples (block S5). A content object is created (block S6) with this content object being stored in temporary data 522.

Build page process 520 uses the URL to select a JSP template from templates 608 (block S7). This template is

Docket No. AUS920010083US1

selected to provide a presentation of the content for a particular content source. In these examples, the type of presentation for a particular content source is associated with a URL. Further, depending on the
5 particular content retrieved from a particular content source, the presentation may vary for different content from that particular content source.

The particular template selected also may be based on particular preferences of the user originating the
10 request. For example, a user may have a preference to avoid intensive graphic animations. A template may be selected for this user in which these types of animations are absent. These preferences may be associated with a user name received in the request. The preferences may
15 be stored within session database 602 in these examples with the user name being used as a query into the database.

A JSP request is created (block S8). This request uses user, navigation, and content objects stored within
20 temporary data 522 to create a JSP. This JSP is returned to the user originating user request 600 (block S9).

In obtaining content in block S5, the URL is used to identify that particular content source. A number of different mechanisms may be used to identify the correct
25 content source from the URL. For example, domain name system (DNS) information at a DNS server may be altered to direct the request to the appropriate content source. DNS is a name resolution system allowing users to locate computers on a UNIX network or the Internet (TCP/IP
30 network) by domain name. The DNS server maintains a database of domain names (host names) and their corresponding IP addresses.

Docket No. AUS920010083US1

Alternatively, the mechanism of the present invention may employ a lookup system. With reference next to **Figure 7**, a diagram of table used to identify content sources is depicted in accordance with a preferred embodiment of the present invention. Table 700 includes entries 702-708 in these examples. Each entry includes a URL, a server identification, and a template. When a URL is identified from a request, this URL is used as an index into table 700 to identify the appropriate source. The server identification may take various forms, such as the name of the server, or an IP address for the server. In the depicted examples, table 700 is implemented within application logic layer 504 in **Figure 5**. Further, table 700 also may be used to identify a template for a particular content source. In this manner, consistent presentation of content may be applied to content sources from the process of the present invention.

Turning next to **Figure 8**, a flowchart of a process used for processing a request for content is depicted in accordance with a preferred embodiment of the present invention. The process illustrated in **Figure 8** may be implemented in a process, such as page creation process 402 in **Figure 4**.

The process begins by receiving a request from a user (step 800). The request is then parsed (step 802). The request is parsed to obtain information, such as, for example, the URL, the source of the request, destination of the request, a user name, and a user password. Next, session information is generated (step 804). This session may be generated from the information obtained from the request and by querying a database of session

Docket No. AUS920010083US1

information. A user object is created using the session information (step 806). Then, the user object is stored (step 808). In these examples, the user object may be stored in temporary information, such as temporary data 5 522 in Figure 5.

Navigation information is obtained using a URL from the request (step 810). The URL may be sent to a navigation database, such as navigation database 508 in Figure 5. The navigation information identifies other 10 pages to which a user may travel from the page identified by the URL. A navigation object is created (step 812) and then stored (step 814). This object also is stored in a temporary location.

Next, content from an appropriate content source is 15 obtained using the URL from the request (step 816). A content object is created (step 818). Next, the content object is stored (step 820). As with the user object and the navigation object, the content object also is stored in a temporary location.

20 Then, using the URL from the request, a template is selected (step 822). A particular domain name within the URL for a content source may be used to select the template. In this manner, a presentation of content may be made consistent for a particular content source. A 25 finer granularity may be implemented in which multiple templates may be associated with a particular content source. Different templates may be selected based on particular content identified in the URL. For example, content for sales information may be associated with one 30 template, while content for technical information may be associated with another template. Further, the particular template may be selected based on particular

Docket No. AUS920010083US1

user preferences for the user originating the request.

A JSP is created using the user object, navigation object, and content object (step 824). These objects are in a format that is usable for creating the JSP. The JSP
5 is then returned to the requester (step 826) with the process terminating thereafter.

Using the mechanism of the present invention, a common service is provided for handling requests for content from multiple content sources. Additionally, the
10 mechanism of the present invention allows for filtering the request to package a response, such as a JSP, that incorporates the original content with a consistent presentation. Further, the mechanism of the present invention avoids requiring developers to modify code each
15 time a content provider changes the content or implementation. The presentation of the information is centralized and separated from the content providers through the use of templates or other similar mechanisms to handle presentation of the content. As a result, the
20 need for content developers to interface with presentation functions is avoided. This mechanism provides scalability because of the ability to handle different content sources. Further, the mechanism allows personalization of the presentation of content.

25 It is important to note that while the present invention has been described in the context of a fully functioning data processing system, those of ordinary skill in the art will appreciate that the processes of the present invention are capable of being distributed in
30 the form of a computer readable medium of instructions and a variety of forms and that the present invention applies equally regardless of the particular type of signal bearing media actually used to carry out the

Docket No. AUS920010083US1

distribution. Examples of computer readable media include recordable-type media, such as a floppy disk, a hard disk drive, a RAM, CD-ROMs, DVD-ROMs, and transmission-type media, such as digital and analog communications links, wired or wireless communications links using transmission forms, such as, for example, radio frequency and light wave transmissions. The computer readable media may take the form of coded formats that are decoded for actual use in a particular data processing system.

The description of the present invention has been presented for purposes of illustration and description, and is not intended to be exhaustive or limited to the invention in the form disclosed. Many modifications and variations will be apparent to those of ordinary skill in the art. The embodiment was chosen and described in order to best explain the principles of the invention, the practical application, and to enable others of ordinary skill in the art to understand the invention for various embodiments with various modifications as are suited to the particular use contemplated.